

REMARKS

This amendment is submitted in response to the Office Action mailed on October 6, 2005. After the response to the preceding Office Action, all rejections of the claims have been withdrawn. However, new rejections of claims 1-5, 11-16 and 23-32 have now been made. In particular, claims 1-5 and 27 are rejected under 35 U.S.C. 103(a) as unpatentable over Duluk, US Patent 6,288,730 (Duluk 1) in view of Duluk, US Patent 6,771,264 (Duluk 2) and further in view of newly cited Bugnion, US Patent 6,704,925. Claims 11, 12, 15, 16, 23, 25, 26 and 28-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 in view of Bugnion. Claim 13 is rejected as unpatentable over Duluk 1 in view of Bugnion and newly cited Wood, US Patent 6,204,856. Claim 14 is rejected under 35 U.S.C. 103(a) as unpatentable over Duluk 1 and Bugnion, in view of newly cited Isard, US Patent Application 2004/0207623. Claim 24 is rejected under 35 U.S.C. 103(a) as unpatentable over Duluk 1 and Bugnion, in view of newly cited Knittel, US Patent 6,266,733.

In this response, Applicant amends claims 1 and 11 to more clearly recite and emphasize the novel features of the present invention. Applicant cancels claim 3, 4 and 5 without conceding the unpatentability of the subject matter thereof, and submits new claims 33-35 to provide a full scope of protection for the submitted invention without exceeding 20 claims.

Conventionally, a shading pipeline, such as a fragment processing pipeline included in the graphics processor, is flushed (using a flush instruction as taught by Duluk 1, Duluk 2 and Bugnion) prior to executing fragment program instructions to avoid any Read After Write (RAW) conflicts. A RAW conflict exists when a write to a position (for example, within a frame buffer or the like) is pending when a read from the same position is ordered. The RAW conflict ends when the write to the position within the buffer or the like is completed.

In various embodiments disclosed and claimed herein, a flush is not needed between writing a buffer and reading a buffer. The necessity of the flush is avoided by performing a conflict check on the destination location, position, pixel, or other object of

the write before, or while, a fragment is processed. If the conflict check indicates that a conflict either exists or is possible, then the processing of that particular fragment is delayed. However, rather than executing a flush, as has previously been done, in this invention, other fragments continue to be executed while the processing of the fragment is delayed. When it is determined that the conflict has been eliminated by a report to the conflict detection unit, the processing of the delayed fragment is completed. In this way, the necessity of constantly providing a flush instruction is eliminated. Further, processing speed is optimized. Such an approach is not taught or suggested by the references cited. Therefore, the rejection of the claims made by the Examiner is respectfully traversed.

At pages 2 and 3 of the Office Action, the Examiner responds to the previous amendment in the subject application. In his reply, the Examiner proposes a broad interpretation of the claims as covering any graphics processor wherein the reordering of operations takes account of conflict detection while eliminating flush instructions. In fact, as clearly presented in previous independent claim 23 and now emphasized in independent claims 1 and 11, the implementation proposed is a fragment processor processing a plurality of fragments, where a first fragment is being processed with the output written to a storage position which is read at some point in the processing of a second fragment. This is the conflict which is detected, and the processing of the second fragment is delayed until completion of processing of the first fragment. The completion of processing of the first fragment is reported to the conflict detection unit and the second fragment resumes processing. This all occurs without the use of flush instructions as conceded by the Examiner.

The Duluk references even when combined fail on a number of accounts. Both the Duluk references utilize flush instructions. More importantly, the Duluk system, especially in the portion cited by the Examiner (column 14, lines 27-40 of Duluk 1) expressly teaches that address requests for data are either forwarded to control block 2605 or in the event of no conflict are placed in a conflict queue 2604. Clearly, the conflicted address will not be accessed and reported until the queue is read out, resulting in a much longer delay in processing of the delayed fragments than occurs in

the claimed system wherein clearance of the conflict is reported/detected and processing of the delayed second fragment then resumes. All this occurs in a graphics processing environment where delay is less tolerable.

The Examiner attempts to remedy this inadequacy by citing Bugnion. However, Bugnion in column 13, line 60 to column 15, line 50, does not teach elimination of flushes. Rather, Bugnion provides an alternate approach which may be adopted under certain circumstances, and these certain circumstances occur (as taught at column 14, line 60) only after a flush of the system occurs and a threshold number of memory traces is reached. Therefore, Bugnion suffers from the same inadequacy as the Duluk references and the combination, if possible (as they are taken from different arts) would still utilize flushes. Moreover, Bugnion, as with the Duluk references, fails to teach delaying the processing of a second fragment only until the processing of a first fragment is completed, and then resuming processing of the second fragment upon the availability of the result of the first fragment process being reported/detected.

The rejection based on Bugnion also relies on the teaching of the translation cache not needing to be flushed: "It suffices to patch the scale translation with a jump to the new translation, thereby guaranteeing that the stalled translation would never again be executed." (See rejection of claims 1 and 11.) This quote clearly contradicts the operation of the claimed invention. If the teaching were followed, then once the execution of the second fragment was stopped, a jump to a new fragment would be executed and the second fragment would be lost and not executed. This would be a fatal flaw in graphics processing if the processing of fragments could be selectively and randomly eliminated and thus never executed.

In his analysis of Duluk 1 in relation to claim 23, the Examiner notes that if a conflict is determined, the conflicting address is sent to the conflict queue. Therefore, the Examiner argues "the processor inherently waits until the address request that is in conflict has finished processing." This is inconsistent with the recitation of the claims. In the claims, the instruction is executed upon report that the execution of a previous

Patent

Attorney Docket No. NVDA/P000814

fragment has been completed rather than waiting for the stalled or halted or locked instruction to execute from a queue.

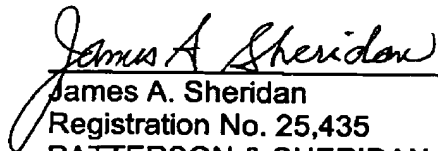
Moreover, as previously presented in claim 24 and now also appearing in claim 35, other fragments are being processed while the second fragment is being delayed without the unlocking of the second fragment, thereby further speeding up the processing. A review of the other references cited by the Examiner does not find any teaching which supplies the deficiencies of the Duluk references or Bugnion relative to this feature.

As to claim 24 (and new claim 35) the Examiner further relies on Knittel. However, this rejection is not proper as Knittel fails to teach anything about locking the processing of fragments in a graphics processor, and then unlocking the locked fragment upon completion of the processing of the first fragment.

In yet another feature of the invention, (as disclosed at application paragraph [0046], and appearing in new claim 34) writes to the buffer can be to coordinate positions which are directly related to coordinate positions in the display. This feature is not disclosed in any of the references cited.

In view of these clear and substantive distinctions between the references cited and the claimed invention, reconsideration and allowance of the pending claims are respectfully requested.

Respectfully submitted,


James A. Sheridan
Registration No. 25,435
PATTERSON & SHERIDAN, LLP
3040 Post Oak Boulevard, Suite 1500
Houston, TX 77056-6582
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Applicants